10 References

- "Milestone 1 Requirements and Architecture, EOSDIS IV&V Infrastructure and Tools", Document NAS5-32605, dated July 15, 1994.

- "Integrated Support Environment (ISE), IV&V System Requirements", Document Draft, dated August 31, 1994.

- "Methodology for the Application of Tools to Independent Verification and Validation", Original Version by Dr. Richard Rogers, undated. Edited version by Mr. Frank Rockwell, dated March 18, 1994.

- "Volume IIB: The Performance Analysis and Technical Evaluation Process", Logicon, document R:SED-80204-IIB, dated June 1980

- "Engineering Discipline for System Development", Computer Sciences Corporation, preliminary draft dated September 5, 1984

- "Software Risk Management through Independent Verification and Validation", Dept of Statistics & Computer Science, WVU, White Paper produced under: DICE program, NASA Grant NAG 5-2129 and NASA Cooperative Agreement NCCW-0040

TBD	To Be Determined
TDF	Tool Development Folder (for purchased tool software)
WAN	Wide Area Network
WVU	West Virginia University

9.2 List of Acronyms/Abbreviations

CASE	Computer Aided Software Engineering
CDRL	Contract Data Requirements List
CM	Configuration Management
COTR	Contracting Officer Technical Representative
COTS	Commercial Off The Shelf
CR	Change Request
DM	Data Management
ECS	EOSDIS Core System
EDOS	EOS Data and Operations System
EO-ICWG	Earth Observations International Coordination Working Group
EOS	Earth Observation System
EOSDIS	EOS Data and Information System
ESDIS	Earth Science Data Information System
GFD	Government Furnished Data
GFE	Government Furnished Equipment
GFS	Government Furnished Software
GOTS	Government Off The Shelf
GSA	General Services Administration
IIR	Integrated Information Repository
IR&D	Independent Research and Development
ISE	Integrated Support Environment
IV&V	Independent Verification and Validation
LAN	Local Area Network
MOU	Memorandum Of Understanding
NASA	National Aeronautics and Space Administration
OO	Object Oriented
PC	Personal Computer
POC	Point Of Contact
SDF	Software Development Folder (for developmental tool software)
SPR	Software Problem Report
STSC	Software Technology Support Center

methodology - A collection of methods, procedures, and standards that defines an integrated synthesis of engineering approaches to the development of a product.

procedure - A written description of a course of action to be taken to perform a given task.

process - A sequence of steps performed for a given purpose; for example, the software development process.

qualification testing - Test conducted as part of the verification program to demonstrate that the design and performance requirements can be realized under specific conditions.

software development plan - The collection of plans that describe the activities to be performed for the software project. It governs the management of the activities performed by the software engineering group for a software project.

user - Any person accessing the EOSDIS. AUTHORIZED USERS are users who have viable EOSDIS accounts, and who may therefore make EOSDIS requests. These users may be affiliated or unaffiliated. This group also includes the: IV&V ISE support staff, IV&V Task Teams, WVU Faculty and Students, NASA Project Personnel, and the EOSDIS Developers. The largest and most important group of users will be the Scientific User Community, the end users. AFFILIATED USERS are those who are sponsored by one of the parties to the Earth Observations ICWG (EO-ICWG) data policy. Each party is responsible for ensuring that all of its affiliated users comply with the EO-ICWG data policy. Use of data by Affiliated Users is classified in one of three categories, defined in the EO-ICWG data policy:

Research Use. A study or an investigation in which the user affirms:

(1) the aim is to establish facts or principles;

(2) the data will not be sold or reproduced or provided to anyone not covered by this or another valid affirmation;

(3) the results of the research will be submitted for publication in the scientific literature; and

(4) detailed results of the research will be provided to the sponsoring spacecraft operator as agreed between the researcher and the sponsoring spacecraft operator.

Environmental Monitoring and Operational Use. Includes data use by those government agencies affiliated with the parties which conduct environmental monitoring and/or operational observations for the public good, and can include larger agencies to which the parties belong or national agencies, or their designates, involved in other operational forecasting activities which are conducted for the public good.

Other Users. Those persons requesting data for scientific, operational, applications, or commercial use, who are not directly represented by an EO-ICWG member, and who agree to the stipulations on data access and use as set by the EO-ICWG and the EOS Program.

verification - The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

9 Glossary and List of Acronyms/Abbreviations

9.1 Glossary

acceptance criteria - The criteria that a system or component must satisfy in order to be accepted by a user, customer, or other authorized entity.

acceptance testing - Formal testing conducted to determine whether or not a system meets specified requirements or criteria and to enable the customer to determine whether or not to accept the system.

configuration control - An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification.

configuration identification - An element of configuration management, consisting of selecting the configuration items for a system and recording their functional and physical characteristics in technical documentation.

configuration management - A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements.

EOS Data and Information System - A facility that manages the data resulting from NASA's Earth science research satellites and field measurement programs, and other data essential for the interpretation of these measurements. It provides access to data held in the archives of other government agencies, organizations, and countries. EOSDIS will generate user-defined data products, and will facilitate the combination and manipulation of data from all sources as well as their incorporation into models of the environment. Concomitant to fulfilling its data management functions, EOSDIS will encourage interdisciplinary research and assist in breaking down the intellectual barriers between the traditional disciplines of Earth science by offering an integrated view of environmental data.

formal review - A formal meeting at which a product is presented to the end user, customer, or other interested parties for comment and approval. It can also be a review of the management and technical activities and of the progress of the project.

integrated software management - The unification and integration of the software engineering and management activities into a coherent defined software process based on the organization's standard software process and related process assets.

metadata - Descriptive information pertaining to data sets. This includes data set descriptions in directories, guides, and inventories, plus any additional information that defines the relationships among them.

8.2 Tool Modification/Upgrade/Substitution

Tools are maintained over time, and the type of maintenance provided depends upon how the tool was acquired. If the tool was purchased as COTS, the vendor usually offers a maintenance agreement which provides periodic upgrades and maintenance releases of the product. End user support is usually provided by the vendor and is sometimes included in the maintenance agreement. New releases may contain new features or improvements that have to be incorporated into the IV&V ISE. As a result, it may be necessary to perform the tool integration process for a new release of an existing tool. Also, changes in the operating environment of the tools, such as new releases of an operating system, sometimes require tool maintenance, due to limited downward compatibility.

Tools developed by the IV&V project are maintained by the developers. Reusability is a prime concern and documentation for each ISE tool is being prepared under established development guidelines. Maintenance will be done on an as needed basis and as dictated through the use of the concurrent feedback process. Due to the long IV&V Program life cycle, the development tool library will continue to grow, becoming very extensive. These developed tools will typically satisfy only a current need for a specific phase of the life cycle. In these cases, once a tool's usefulness is passed, the tool will be shelved within the IIR for potential reuse later in the project.

When the tools are shared tools from outside the ISE, policies and procedures will be established for the use of the tools. Issues such as information security, data integrity, and utilization coordination are addressed. Since the IV&V project does not own these tools, or the information within the tools, close communication with tool maintenance personnel is essential to avoid incorrect use of the tools.

As a result of user feedback, enhancements or deficiencies will drive changes to the tool suite within the ISE. When a developed tool is unable to respond to these changes, the tool selection process is invoked to initiate the search for a tool that will now meet the new requirements. The feedback and subsequent tool selection process will identify whether a new tool is developed, a shell is written around the old tool, or a shell is generated around another tool to promote reusability. Performance/validation testing is performed subsequent to any development activity, and using the old tool as the kernel for new plans and procedures would realize a significant time saving. However, if a COTS tool is the answer due to possible cost and schedule impacts, then the procurement process is initiated followed by acceptance testing using the new updated tool requirements.

8 Life Cycle Maintenance

Care is taken to ensure proper and timely tool execution, data management, and accurate analysis and distribution of results. All tools are maintained and upgraded under configuration control. Changes in developer's work products and processes, refocusing of the IV&V effort, or changes to the IV&V scope, may levy new requirements on the toolset used by the IV&V analysts. Improvements to existing tools may be identified to improve IV&V results, or tools of benefit to the project may become available from external sources. In all cases, there is a constant evaluation of IV&V tool effectiveness by the generation of user indices, and a concurrent effort to continuously improve on the operational tool suite by the solicitation of improvement recommendations to maximize the pay back on tool investments.

Through the use of a concurrent feedback process, changes to the existing tool set will be mandated. Any and all revisions and/or upgrades to the tools are validated in the same manner as indicated in the previous sections. As requirements change, for any number of reasons, or more efficient tools appear, the Tool Acquisition, Development, and Maintenance cycle is repeated to ensure a consistent tool set evaluation, targeted to meet the needs of the IV&V analysts.

8.1 Tool Maintenance Improvement Process

The objective of process improvement is to learn from program experiences and improve upon the methods and tools that are beneficial and discard the ones that are inefficient. This is the primary reason it is important to monitor and improve its processes. Toward that end, a process improvement plan is incorporated. The plan includes information exchange forums, product reviews, project post-mortem evaluations, and IR&D activities.

One mechanism that is vital to an IV&V Program is the information exchange forum. By exchanging information between users, such as product reviews and information, a program is able to share valuable experience across many projects and tasks. Post-mortem analysis, where the project is evaluated from both positive and negative aspects, also contributes to a body of information that is used to improve the overall IV&V process. New technology and process improvement research is supported through the implementation of IR&D activities. A reassessment of requirements and user needs is constantly in process to insure that all tools are responsive. If indications are that user dissatisfaction is on the rise, with concurrent technology evaluation and IR&D activities, the EOSDIS IV&V team is ready to be responsive to the changing needs of the user community.

Through Lotus, the IV&V team has established a Notes Database to accumulate user feedback. This feedback is received as phone, E-mail and mail, as well as Local Area Network (LAN) and Wide Area Network (WAN) user access. The target database that receives the feedback has several "views" compliant with each particular application. A report is generated categorizing this information by various fields (e.g., source of feedback, specific tool, process addressed, generic recommendations). This report is used to direct tool software enhancements and process improvement activities.

location, site directions, and course content. Whenever a course is completed, a copy of the training materials is made and kept on file for future reference.

Tool training has two aspects: 1) when a new individual joins the IV&V staff, and 2) when a new tool is acquired. When a new individual joins the staff, training is accomplished through several mechanisms, including "hands-on" training, and reference material. The individual is trained in the IV&V methodology and the core set of tools. When the individual is assigned to a particular task, "hands-on" experience with the tools is provided. The tool repository contains reference material for the individual to use.

When a new tool is acquired by the IV&V staff, training is accomplished through courses or seminars in how to use the tool. These seminars usually underscore the methodology that is used in conjunction with the tool to accomplish the tools objectives. The training re-enforces the IV&V methodology while providing "hands-on" experience for the IV&V staff.

7.2 Tool Implementation/Utilization

The goal of tool configuration management, maintenance, and support is to provide a set of tools that operate efficiently and protect the toolset as a NASA IV&V asset. It is important to focus the ownership of the tool repository on an established function within the project. It is the responsibility of the tool repository function to make sure the tools operate properly, the users have the latest releases, and deficiencies are reported and fixed. Tool library and repository management provide a focal point and mechanism for tool maintenance.

Operation is one of the last steps in this process and a remaining issue to be addressed is the periodic maintenance that automated tools require, such as procedures for periodic protection of the information base and other routine maintenance. Server backups are scheduled to be performed on a routine basis. An 8mm tape backup device is used for all backups. Monthly backups will be total, while differential backups will occur daily. Backups for 6 months will be kept, and then the tapes will be erased and reused.

The IV&V facility has a Uninterruptable Power Supply (UPS) to prevent accidental power loss to the servers and processor units on the ground floor. The second floor is protected by individual UPS units assigned to each IV&V processor to allow graceful degradation of any process currently running.

7 Training and Operation

All IV&V resources are trained in tool operations. Vendors are contacted to train the IV&V engineers, as required, in the use of their product, and hands-on demonstrations are provided when deemed necessary. Supervision early in the implementation period aids in ensuring productive and efficient use of all IV&V tools.

7.1 User Training

Part of effective IV&V tool education is providing a strong theoretical background as well as a practical background. Methodology training provides the theoretical background, while tool training provides a more practical application of the methodology.

The information resources available to IV&V contain literature with information on new methods and techniques. The IIR is used to train new analysts and supplement the knowledge of the current staff. Some of this information is contained in an IV&V policies and procedures handbook that new members of the project receive. Along with the handbook, new members of the project use the repository to become familiar with the IV&V methodology, the tool methodology, and the tool infrastructure.

Training in IV&V methodology and a strong understanding of how tools are applied to the methodology are important fundamentals in using any IV&V tool. To maximize the effectiveness a tool, users need to understand:

- The importance of the methodology supporting the tool,
- The source of the information managed by the tool,
- Its significance in the system development life cycle, and
- Its relationship to other information gathered by the IV&V effort.

Tool release to the project toolset is a formality accomplished after successful completion of testing. An instance of the tool and necessary support aids from the information repository (e.g., training materials, operations manuals) are supplied to the IV&V project at this milestone.

Tool training is necessary to ensure the staff understands the features of the tool and its potential value to the project. Training needs are determined by polling the Program Manager and Task Leaders on a quarterly basis, receiving recommendations from team members on an ad hoc basis, and obtaining direction from the NASA COTR. Once a training need is identified, various program alternatives are considered, including prepackaged seminars, academic courses, self-paced tutorials, and tailored presentations. A comparison is made between the curriculum offering and the program cost, and alternatives are discussed with the relevant managers. When a specific training program is agreed to, it is scheduled to provide minimal impact on contract performance. A written announcement of the course is issued to all participants indicating time,

6.2 Acceptance/Validation Testing

Acceptance/performance is conducted incrementally, one tool at a time, starting with those tools that are being reused from the tool infrastructure. Each such tool is imported to the ISE configuration and tested for performance against project requirements. Tools coming from outside are first installed in the ISE, and then tested for performance in their project configuration. Once testing of a tool is complete, the tool is released to the project toolset.

For tools that were originally part of the ISE and are being targeted for reuse on the project, this testing is commonly referred to as Validation Testing. This testing ensures that the shared tools are compliant with user needs and satisfy requirements. For tools purchased and brought in from external sources, this testing fulfills the Acceptance Test milestone. In either case, tool testing is directed towards demonstrating that the performance requirements documented as part of the needs assessment are satisfied. For tools that are reused/modified or developed, testing focuses on performance aspects to ensure that the tool functions correctly, and satisfies intended objectives. In this case performance testing is able to test at a lower level since the components are visible; whereas, when something is imported as an entity, this visibility is not present.

6.3 Tool Validation and Post Test Analysis

The validation process occurs after the tool is integrated, controlled, and tested, but before the tool is placed into the ISE as an operational element. As part of the validation process, the tool is operated concurrently with a manual process, and evaluated for process and information accuracy. If, for example, documents were being corrupted because of loose configuration control, the project might re-examine and modify the process, and/or tool, to impose more strict control.

DETECTION INFORMATION
- Data regarding how the problem was detected and the environment where the problem was found.
- Problem severity, project affected by this problem, and date fix is needed
- Type of problem, e.g., enhancement, show stopper, defect, annoyance

SUBMITTER INFORMATION
- Describes who submitted problem, including project, phone, e-mail/address and date

LABORATORY INFORMATION
- Data relating when problem was evaluated/opened and date assigned to engineer
- Field noting type of problem and recommended change
- Analysis time, estimated time to fix, and date resolved

RESOLUTION INFORMATION
- Information on how problem was resolved, by whom (name, project, etc.), actual time to fix, and date resolved

VERIFICATION INFORMATION
- Identifies who verified/approved problem resolution (i.e., name, etc.), and date SPR was approved

ENCLOSURES
- Problem description entered by submitter should contain detailed description of the problem.
- Indication of units/documents affected, also entered by submitter, should contain a list of elements affected by the problem

The CR process will be entirely contained within the Configuration Management software tool. CR forms will be filled out electronically with reference to the parent SPR and stored to facilitate tracking and reference of the particular CR with any version activity resulting from software modifications. This will all be fed back to the parent SPR for closure.

Configuration Status Accounting provides essential project management information that relates the status of change processing and implementation actions to established releases. Configuration status reporting takes the form of a monthly status report and includes, at a minimum:

- Current schedule,
- Progress indication,
- Summary of current changes including:
  - Reason for change and
  - Impact of change, and
- Definition of key milestones and deliverables.

As an adjunct to configuration status accounting, an underlying software review activity coordinates the problem reporting system. SPRs may be originated by anyone on the project or using the tool. This activity is tasked with reviewing and categorizing them as system, hardware, or software. The SPRs are then assigned to a specific team for corrective action. The CM/DM Library maintains the master file of all SPRs and periodically publishes a current index of all reports.

the software, or is reported to the tool vendor. Vendor problems can become schedule critical due to response time if a software fix is recommended for vendor software.

If the tool was developed in house, the next step is to analyze the SPR to resolve the problem. At this point, several SPRs may be collected to initiate a Change Request, or if the problem is a critical or severe problem, spawn its own CR. After the problem is resolved and the fix is completed, regression testing is performed to revalidate a new version of the tool software. Occasionally, new test procedures are generated since the problem was not revealed during previous development testing. All applicable documentation affected by the change is revised and republished. The final step is to close the SPR and report to the submitter.

The format of the SPR provides fields to facilitate various searches and report generations. The basic categories of information included on an SPR form are listed below, and an example of a SPR is illustrated in Exhibit A.6-7.

Software Problem Report

Incident #: AAAbb99999

Title:

Source:

	NAME	PROJECT	PHONE	DATE
Submitted By:				
Resolved By:				
Approved By:				

Review Block:	NAME	PROJECT	PHONE	DATE
Closed By:				

Problem Description:

Software/Documentation Affected:

Recommended Solution:

Actual Solution:

Problem Category:
Requirement Clarification _____ Interface Error _____ Error Logic _____
Requirement Change _____ Standards Error _____ Other _____
SPR 9/30/94

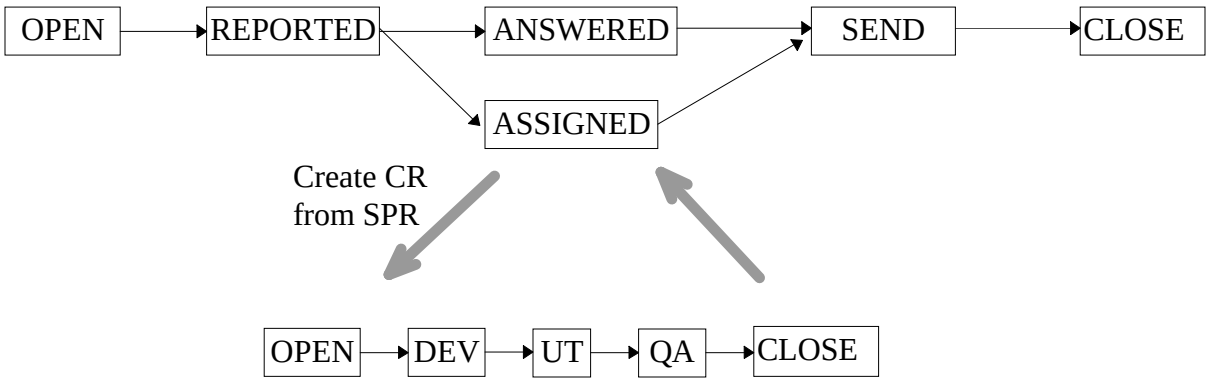
EXHIBIT A.6-7. Example of SPR

IDENTIFICATION INFORMATION

- A log number assigned by the change management software pending approval by lead analyst
- Title annotated by submitter, incorporating a short description of the contents including the project name
- Identification of source program or document in which the problem was discovered, including version number

incident is validated to determine its credibility. If the incident is considered valid, a log number is assigned to the incident and it becomes an SPR.

Life cycle for a Software Problem Report (SPR)



Life cycle for a Change Request (CR)

EXHIBIT A.6-5. Relationship of SPR and CR Life cycles

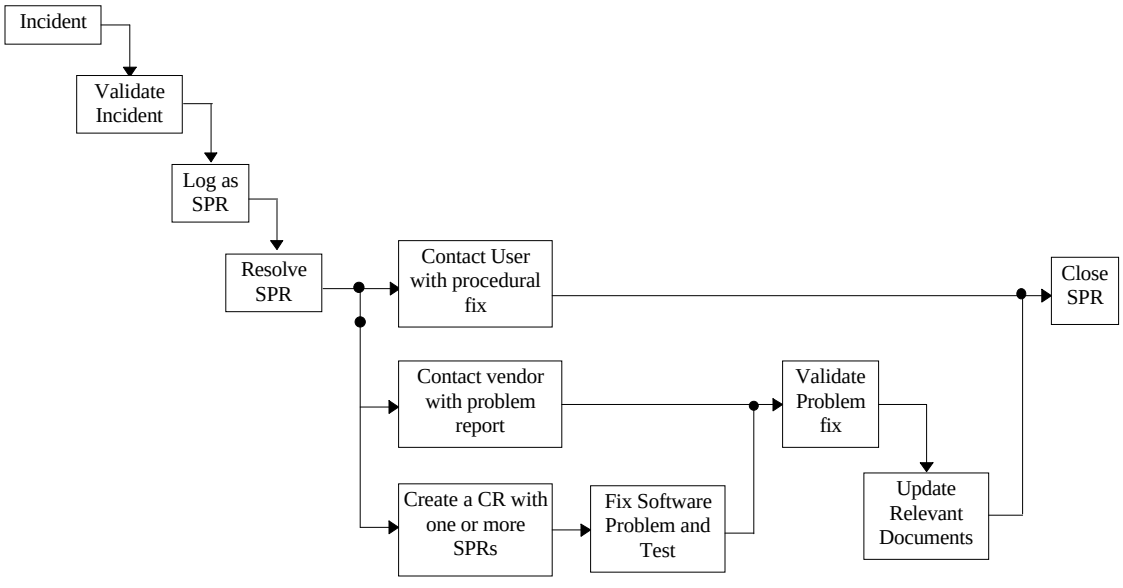


EXHIBIT A.6-6. IV&V SPR Process

Several alternatives exist depending upon the nature of the problem. The lead analyst may determine that the problem is incidental and requires only a discussion with the submitter (i.e., operational problem and procedural work around is adequate), actually requires a modification to

The CM process is initiated by the baseline identification of configuration objects. It is the process of establishing, documenting, and baselining the requirements for, and configuration of, all software tools and products over the development life cycle. The objects explicitly tied to the tool are also identified via this process. In addition, other software life cycle objects, e.g., System Development Plans, System Test Plans, are also controlled through this CM Activity. Products entering the system from outside the environment and not directly involved with a specific developed software tool or life cycle are excluded in this CM process and are handled by the DM activity.

Key to implementing configuration management is the correct identification of the point at which an item is placed under configuration control. Placing the object under control too early creates the administrative burden of tracking changes before the item has stabilized. Placing the object under control too late permits uncontrolled changes, allows different versions of the object to be used, and subverts quality control activities. All tool software will be placed under configuration control when it is submitted for formal testing. This is the point where the tool software is prepared for controls to be imposed on it. Tool documentation is placed under formal control prior to the initial formal review of the object by NASA. When formal NASA approval is not required, the point for instituting formal control will be 30 days after delivery. At that time NASA is formally advised of this action.

Changes to software or documentation are initiated as a result of software problems, documentation errors, or changes in requirements. Positive control is established to preclude unauthorized changes to an established baseline. There are procedures to ensure that each proposed change to the baseline is completely described (including impacts), is thoroughly coordinated, reviewed, and evaluated. These ensure that changes are authorized and implemented in an approved manner. The ISE will only accept a baseline change that has been processed and approved in this prescribed manner.

Changes to identified configuration objects are managed through the use of an informal IV&V Configuration Control Board. All proposed changes to an established baseline object are documented, evaluated, coordinated, and dispositioned. A Software Problem Report (SPR) begins its life cycle as an incident. After the incident is reported, it moves to the resolution phase where it is dispositioned. This step involves review of the SPR and may only include a response to the submitter to update their procedures for closure, or it may entail assignment to a software engineer for analysis, implementation, and eventual response to submitter for closure. In the latter case, an incident may be severe or frequent enough to spawn a Change Request (CR). At this time the assignment process will initiate a CR activity. The CR life cycle passes through a Development Activity, Unit Test Activity, and a software review activity before returning to the SPR life cycle, to be sent to submitter and closed. Exhibit A.6-5 illustrates the relationship between these two life cycles.

An SPR process is being implemented by the EOSDIS IV&V Team. The IV&V Tool SPR process is tailored specifically for ISE problem resolution. Exhibit A.6-6 illustrates how this process is performed. Initially a problem enters the system as an unsubstantiated incident. The

Electronic Notebook Check-out Card

DM Control Number

NAME	DATE

EXHIBIT A.6-4. Notebook Checkout Card

tool used to implement the CM process will be COTS, with several candidates under consideration. A separate, but compatible CM tool will be hosted on the PC to support tool development on the PC and Sun platforms. A more capable and robust CM tool will be used to provide the CM capabilities during IV&V testing of EOSDIS components, interfaces and systems. The selection of this tool hinges upon the tool(s) chosen by the EOSDIS developers because electronic compatibility is desired.

Requirements levied on the CM process include the need for configuration identification, configuration control (version control), status monitoring and the use of reviews and audits to support the CM process. Processes include the management of developmental configurations, establishing software version baselines, and identifying and controlling software changes for the tool set targeted for the ISE. These processes break down into explicit activities inherent within this Configuration Management control process:

- Configuration Identification - Definition and documentation of deliverable end-item products,
- Configuration Control - Systematic evaluation, coordination, and approval or disapproval of proposed changes to any baseline,
- Configuration Status Accounting - Recording and monitoring of all changes to established baselines, and
- Configuration Audits for Authentication - Formal comparison of deliverable documentation against deliverable configuration items.

The following paragraphs will outline how these activities apply to EOSDIS IV&V tool management.

EOSDIS IV&V EQUIPMENT LOG FORM

Equipment Identification

Item Ctrl #: *(YYMMDD.##, where ## is daily sequence #)*

Item Description:

Date Received: Received By:

Hardware Serial Number: Software Version Number:

Vendor Name: Vendor Address:

Phone Number:

Contact:

Maintenance Agreement: ◯ NORMAL ◯ EXTENDED

Maintenance Agreement Start Date: End Date :

Comments/Warranty Information:

EXHIBIT A.6-3. Equipment Item Entry Log Form

EOSDIS IV&V CARD CATALOG

Document Identification

Primary Ctrl #: *(YYMMDD.##)*

Document Title:

Version: Date:

Prepared By: For:

Document Type:

Responsible ISyS Task:

Primary System(s) Addressed:

Hard Copy: ◯ Yes ◯ No

Electronic Copy : ◯ Yes ◯ No

Location: Hard Copy:

Soft Copy:

Comments:

EXHIBIT A.6-2. Card Catalog Entry Log Form

Once released, the tool configuration is strictly maintained by project personnel. Any changes or modifications to a tool are handled in accordance with procedures established by this document. CM and DM work together and are symbiotic processes. CM catches all of the objects, and some of them are sent to DM who distributes them. CM controls all the objects, and DM verifies and reports on them. The two processes work together throughout the software life cycle.

6.1.1 Data Management Control

The goals of data management are:

- To manage the distribution, storage, and use of vendor supplied or GFD/GFE, and
- To properly track the delivery and status of both government and IV&V team generated software and documents against the Contract Data Requirements List (CDRL).

A master IV&V library has been established to hold documents used and/or generated in the performance of IV&V tasks. The documents used include documents generated and distributed by the EOSDIS Project and the development contractors. Documents generated include IV&V CDRL items, comments on reviewed documents, technical reports, status reports, and study reports/white papers. The physical library containing all hard copy objects is housed at the Greenbelt, MD IV&V facility. The electronic library is distributed between the Greenbelt, MD and the Fairmont, WV IV&V facilities. The Librarian resides in Greenbelt, and is the manager of all hard copy and electronic entries in the library. Users have read only access to the library.

Data Management Control is accomplished through manual entry on customized forms generated within the Lotus Notes environment. These forms contain more than adequate information to categorize all objects into various applicable groups. The forms are able to handle any and all DM activities associated with non-version control type objects, e.g., GOTS, COTS, EOSDIS Documents, miscellaneous publications, hardware. An example of the EOSDIS IV&V Card Catalog Entry Log Form is presented as Exhibit A.6-2. This Lotus Notes form is used to check out hard copies and electronic copies of any documents in the library, hard copy reside in Greenbelt, and electronic copy are distributed. A user may check out any number of documents. An example of the hardware and software Entry Log Form is presented as Exhibit A.6-3. This form is used for the entry of all hardware and software received by the project. Exhibit A.6-4 presents an example of an electronic checkout form for hard copy documentation. This form will be used by the DM librarian to log checkout activity and provide users with visibility of who has limited copy documents.

6.1.2 Configuration Management Control

The goal of CM is to manage, through version control, the configuration of the tools being developed for use on the EOSDIS IV&V Program. As discussed in Section 5.1, these tools pass through the same developmental cycle as a typical software development program. Control of the tool configuration, testing of the tools prior to release, archiving of all releases, and monitoring of deficiency reports and/or changes are all required CM functional capabilities. The

6.1 Configuration and Data Management Control

Two aspects to management control of the numerous objects produced, retained, or received by the EOSDIS IV&V Project Team, are CM and Data Management (DM) Control. Most of these objects are directly related to activities centered around the ISE. An exhaustive list of the products to be placed under CM/DM Control is as follows:

- Commercial-Off-The-Shelf Tool Software (COTS)
  - Load Module
  - Documentation
  - Acceptance Test (input and output data)
  - Tool Folder
- GOTS Tool Software
  - Load Module
  - Documentation
  - Acceptance Test (input and output data)
  - Tool Folder
- In-House Developed Tool Software
  - Source Code (including Libraries)
  - Load Modules
  - All Software Development Documentation (e.g., Manuals, Tool Development Folders)
  - Performance Test (input and output data)
  - Tool Folder (hard copy)
- Shared Tool Software
  - Load Modules
  - All Software Development Documentation (e.g., Manuals, Software Development Folders)
  - Performance Test (input and output data)
  - Tool Folder (hard copy)
- Tool Selection Repository
  - Tool Folders for All Initial Tool Candidates
- User Test Scripts/Input Data
- User Test Validation/Output Data
- Tool System Documentation
  - Software Development Plans
  - Test Plans/Procedures
  - Certification Plans
  - Miscellaneous Reports
  - Requirement Specifications
- Government Hardware
- ISE Hardware

6 Tool Configuration Management (CM) Process

The tools comprising the IV&V tool suite, those acquired from COTS, GOTS, WVU, public domain, or in-house sources, are placed under CM control prior to installation into the ISE. All tools are subject to acceptance testing to validate that the tools function as required and produce the correct results in the intended environment. A folder for each tool under CM intended for integration is maintained, and includes all information relating to its compliance with user requirements and criteria. All tool testing results are analyzed to verify compliance with the initial tool requirements.

Exhibit A.6-1 illustrates the general procedures for configuration and data control for tool management purposes on the EOSDIS IV&V Program. Major functions include data and configuration item logging and tracking, integrity maintenance, distribution, and status accounting. Primary outputs from the configuration and data control systems are CM Status Reports and the Data Accession List. The former summarizes the current status of items under formal configuration control, and the latter details items added to the library/inventory during a reporting period.

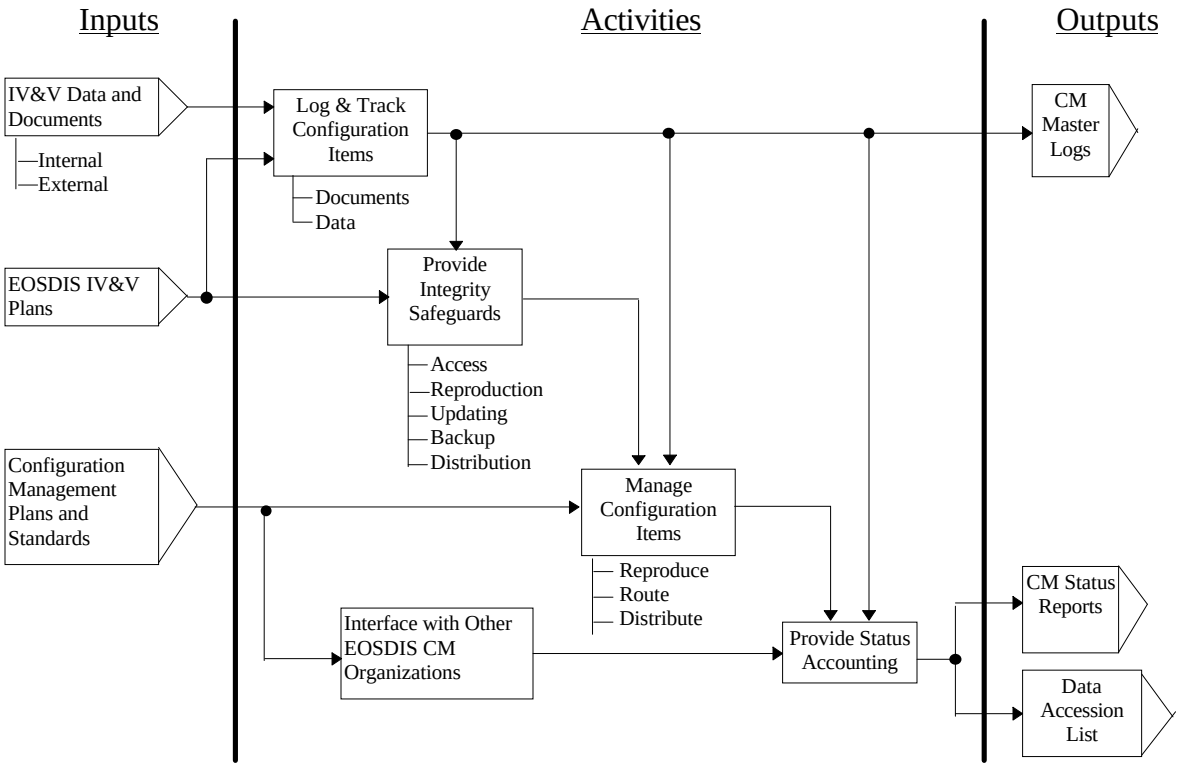


EXHIBIT A.6-1. Data and Configuration Control

In the event test results do not meet the established acceptance criteria, the test engineer is required to notify the development team of the deficiency by submitting a Tool Deficiency Report. If the acceptance criteria for all tests have been met, test results are documented and entered in the respective Tool Development Folder. A test report is then prepared for each tool tested, a copy of which is included in the Tool Development Folder.

Tool deficiency reports result in one of several actions. First, the test procedures may need to be rerun due to improper execution of procedures, or they may need to be modified within the test plan guidelines. Second, the requirements may not have been correctly interpreted or the tool has a bug. In this instance, the requirements are restated and the test procedures changed to reflect a correct interpretation, or, the tool reenters unit test to isolate the problem. Essentially, the failure of a tool to meet its requirements means the tool is inadequate or the requirements are impractical. In some cases, specific criteria may be too stringent and require relaxation. A number of decisions need to be made in regards to a failure of a tool at this point. If the final decision is to reject/eliminate the tool, the tool selection process must be reinitiated, and the entire tool selection and testing process repeated.

5.3 Test Plans & Procedures

All software tools incorporated into the ISE are tested to establish functionality and ensure ISE integrity. This section addresses the need for all tools, developed or non-developmental, to be tested by the integrators according to a tailored set of test plans and testing procedures. Developed tools, built or reused, will have already gone through component level testing and are now subject to Configuration Management Control and ready to be formally tested at the top level, performance testing. Non-developmental tools, bought or shared, will have passed through Data Management Control, and are ready for acceptance testing. Both testing categories form the basis of validation testing for the ISE toolset.

The Test Plan identifies the requirements to be verified, i.e., those initially defined during the tool selection process. A Test Plan is prepared for every tool in the ISE infrastructure. Plans vary in detail and complexity depending on the requirements imposed on the tool. The Test Plan identifies support software, computers and other hardware necessary to stimulate the tool, and to collect data for subsequent analysis and reduction.

Test Plans define the test objectives, stimuli, and expected responses for developed tools. Tests chosen for the Test Plan often represent a subset of tests run at lower levels during unit testing. These final tests are selected from among the lower level tests and are based on complying with the following criteria:

- The ability of the test to verify one or more of the requirements identified previously, and
- The ability of the test to execute without instrumentation for collecting internal data.

For each test selected, the tester identifies expected outputs and acceptance criteria based on information contained in the requirements.

Test Procedures define the test environment, consistent with the environment requirements in the Test Plan. If a basic test environment can be defined that satisfies a set of related tests, then that environment need only be defined once with deltas to the basic environment. Each procedure defines detailed unambiguous steps, to ensure test repeatability, and provide a level of detail such that anyone familiar with the tool would be able to perform them. The procedure, at a minimum, identifies the steps necessary to:

- Start up the test environment and initialize the software under test,
- Maintain the operation of the environment during test,
- Provide the necessary stimuli in the appropriate time sequence, and
- Shut down the test environment and software under test.

Actual tool testing is performed in accordance with the test procedures, and all data collected and recorded are included in the Tool Development Folder. Data reduction or analysis is performed as necessary to determine whether or not all functional and performance requirements allocated to the tool have been satisfied.

5.2 Tool Environment Preparation and Integration Plan

The tool integration process has the objective of creating an effective project toolset, i.e., the integrated set of tools being used to support the EOSDIS IV&V Project. Tools can be incorporated or reused from within the ISE, and also brought in from outside. Tools brought in from outside the domain are initially installed and brought on-line.

When the tool acquisition strategy is to modify or buy, the tools are installed within the tool infrastructure as they are received. This installation creates entries in the ISE, and possibly validates the compile and execution profiles of the tool if software rights are included. Subsequent tool use is regulated and controlled by configuration management procedures within the infrastructure (see Section 6). The major milestone to be satisfied prior to release to the project is that of tool testing (either validation or acceptance).

An integration plan addresses the logistics of using a tool, which includes tailoring a tool to fit the processes defined by this IV&V Tool Management Plan, and also tailoring a tool to fit into the project's computing environment. An integration plan is developed after the set of tools have been identified, and addresses such issues as tool integration, tool validation, and tool operation.

Initially, issues such as how is the tool going to be used, who on the project uses the tool, who provides information for the tool, and who consumes information from the tool need to be addressed. Some of this information is outlined in this IV&V Tool Management Plan. In a situation where the tool is developed, these issues become requirements. Tool software normally provides options which allow tailoring to the user's specific needs. These options are reviewed and implemented as they apply to the user's needs. This involves understanding the tool objectives and tailoring the tool to meet project requirements.

After resolving issues regarding the tool itself, issues such as how the tool interacts with other tools, how the tool communicates with other machines and other sites, and how the tool is installed and maintained, are addressed. Issues implicit in this scenario are those concerns pertaining to tool compatibility with not only local/in-house but customer and applicable corporate data base platforms. To simplify the integration process, it is important to choose a set of tools that are easily ported to the target hardware and software platform. A mature IV&V project has a core set of portable tools. It is important for this core tool set to communicate via a recognized standard. For instance, many CASE tool vendors have developed standards and environments that allow tools to exchange information. The goal is to facilitate the exchange of information and simplify the integration process. Tools selected should be able to take advantage of already established communication facilities and their imposed standards.

Following these guidelines enables a tool to be reused while also facilitating the integration of the tool into other projects. It lays the foundation for creating a tool repository where all tools interact and exchange information.

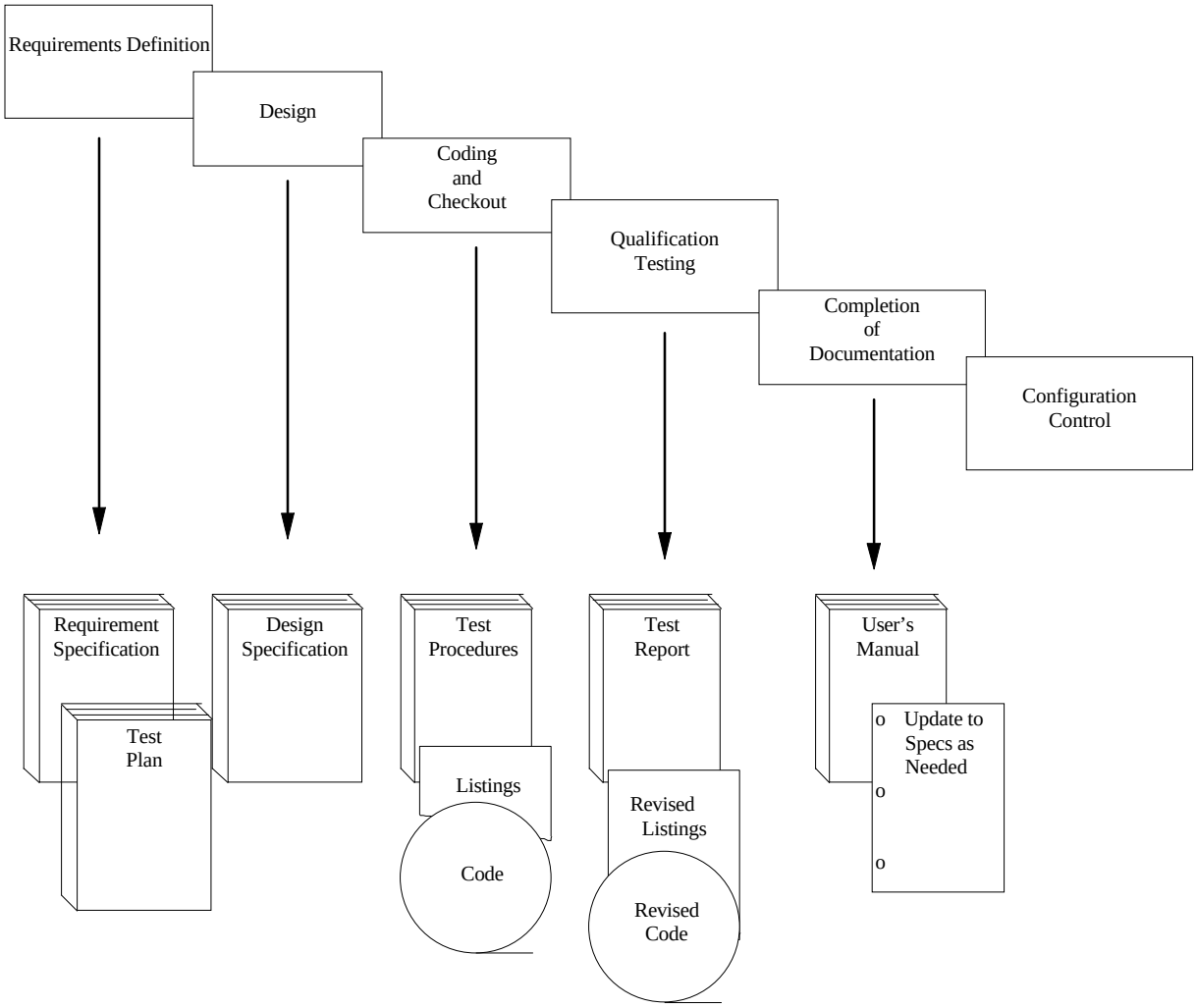


EXHIBIT A.5-1. Tool Development Process

Tool modification efforts depend on whether or not the original tool is maintained as a separate version. If it is, the code and documentation are left intact, and new materials are prepared for the modified version. If it is not, the code and documentation are simply updated. In either case, the development process parallels that for a new development, at each step taking into account the work already accomplished, and the impact of the proposed modifications.

The formality level of tool documentation is dependent upon NASA project office requirements, available resources, and the application domain. All documentation adheres to NASA standards or follows more informal formats depending upon the level of usage, criticality and tool reuse. Regardless of format, documentation provides a complete description of the tool, its use, and the testing used to qualify it.

5 Tool Development/Procurement Process

This process occurs after a buy/share or make/modify decision has been made for a specific tool. If make/modify is chosen, a Tool Development Plan is generated, and the tool is developed or modified in accordance with NASA standard NASA-STD-2100-9191, and this EOSDIS IV&V Tool Management Plan and the specific Tool Development Plan. All supporting documentation relevant to the insertion of this tool into the ISE is included in the tool folder. Support from the IV&V tool infrastructure provides resources for readying the tool environment for integration, development of tool test plans and procedures, and guidelines for acceptance and performance testing.

5.1 Tool Development/Procurement Plans

Tool procurement is straightforward - requirements and needs are documented and the solution identified. The software is ordered through approved corporate purchasing channels. Tool procurement entails purchasing a tool as COTS, ordering a tool as GOTS, or making arrangements for borrowing/sharing a tool. The latter usually takes the form of a Memorandum Of Understanding (MOU) that clearly delineates each party's responsibilities. Procured tools are used, as supplied by the manufacturer, with no modifications. Once the software arrives, data management control and tool integration are initiated.

The development process for each tool depends upon its status at the beginning of the effort. Some tools developed for previous efforts may be usable exactly as they are, hence reused, and others may require modification to be suitable for current effort, e.g., modified tools. Still others may require new development, but, in all cases, each tool is qualified and placed under management control before being applied to the project. In the case of reuse without modification, data management control suffices. However, if tool software is developed, or is intended for modification, it is placed under strict configuration management control to permit baselining and version tracking.

The order in which the tools are developed depends upon the order in which they are needed and the length of time required. Tool development is scheduled so each tool is ready when needed. Developing a new tool entails the same activities performed in normal software development and produces the same suite of documentation as shown in Exhibit A.5-1. The requirements for the tool are identified and specified in the Requirements Specification.

The testing approach is set forth in Test Plans and Procedures. The tool is designed, and the proposed design is documented in a Design Specification. After coding and initial checkout, the tool undergoes rigorous qualification testing. Test procedures specify the testing to be performed and the criteria to be used for tool acceptance. Particular attention is given to ensuring that the tool does not do anything that could mask the presence of the types of problems it is designed to help detect. A test report documents the results of the qualification testing. Updates to the requirements and design specifications reflect any project changes made during coding and testing, and a user's manual describes the tool capabilities, limitations, and usage.

At this point in the tool selection process, candidate tool(s) in each category have been thoroughly evaluated for capability. Management will make a commitment, based upon the contents of the tool folders and the recommendation letters prepared by the tool identification team leader.

The decision to buy/share versus make/modify is by no means a simple matter of how much does it cost and how long does it take, even if the various tools in a given category of tools are comparable. For instance, development cost data is carefully weighed to assess validity, because development costs can conceal expenses until it becomes too late to modify the development approach.

There are numerous subtle factors that are attendant to rendering a decision. Developing a tool will answer 'current' requirements, but when modifications are necessary, the development approach taken may not accommodate change, leaving the team with an inadequate tool. Development is only used when it is not practical to buy or share.

All activities performed during the cost and schedule analysis are detailed within the tool folders. A matrix presenting the results of this study is prepared for submittal to management. This matrix summarizes findings with respect to all of the relevant criteria employed in the decision process.

4.2 Tool Evaluation Exercise

After the tools have been evaluated for value (e.g., cost versus performance), and, after any constraints relating to schedule impacts (e.g., vendor delivery delays, development time lines) have been tabulated, the tool recommendation list becomes shorter and, therefore, more manageable. The next step for tools recommended for procurement is a hands-on evaluation. The procedures require the team POC for each tool to contact the remaining tool vendors, subsequent to the initial screening in the Tool Identification Activity, and acquire evaluation copies or demo discs. To expedite this activity, the vendor may be asked to supply an evaluation copy (this can be accomplished at the Tool Demonstration, Section 3.3, to avoid further delay). Once the POC receives the tool evaluation copies, the following data are compiled: the tool data sheets and the compliance check-off with user requirements and the user "non-imperative" criteria. All notes and comments are captured in tool development folders and provided to the team POC for collation and preparation of final tool recommendations. The POC should have several complete tool development folders that contain the following information:

- Completed criteria checklists,
- Completed tool data sheets, and
- Any notes and comments taken or received during tool demonstrations/evaluations.

4.3 Final Tool Recommendations

The tool identification team lead gathers the files for a given category of tools pending recommendation. The lead examines the tool development folders for each tool in the category and weighs the scores with the numerous comments and notes from the evaluation phase. From this analysis the lead develops a formal recommendation to provide to management. This task is not easy since a formal recommendation memorandum is prepared to justify the purchase of each tool, including the lead's rationalization for recommending a specific tool. The tool folders that are not included in the final recommendation are filed, so that others may draw upon the knowledge and insight that was gained by the tool identification team. The selected tool set recommendation letters and the tool development folders are passed on to management for concurrence. The folders for tools not included in the recommendation may be needed for tool reevaluation resulting form changes in user requirements or failure to obtain concurrence from management for tool procurement or development.

4.4 Tool Selection by Management

tools is that the marketplace may not have tools that satisfy all requirements, or the tools that satisfy the requirements do not share information easily. In some cases, user needs can be recommended to the vendor, who may incorporate them in forthcoming releases, making a better fit with the user's original requirements.

In order to improve efficiency and communication, e.g., information transfer, it may be necessary or advantageous for the IV&V team to use development contractor tools. In this situation the development contractor would be sharing both his information and his tools with the IV&V team. One benefit is that the project as a whole is able to maximize its investment in tools, although it does introduce some logistical problems, such as tool scheduling and utilization. On the other hand, a drawback is that using the same tool for the same purpose introduces an independence limitation. Many occurrences have been documented where implementing a different viewpoint or approach to the same result reveals defects or limitations in the current design.

The tool selection process is a methodological approach to analyzing needs and performing a search for the compliant suite of tools, each of which addresses all, or a majority of the user's requirements. This search or selection process includes hands-on evaluation of products resulting in a final list of tools submitted to management for evaluation when the Buy/Share Tool option is chosen. Management views each tool from a program level prospective, rendering a decision based initially on cost and schedule impacts and constraints, and inevitably on corporate goals and management trends.

4.1 Cost Benefit/Schedule Constraints Analysis

Budgetary limits, cost constraints, and project schedule needs are key factors in the tool selection process. These result in requesting a specific tool from a specific source with an identified budget and availability. Although costs used throughout the tool acquisition process include those associated with use of the tool (an inexpensive tool that consumes inordinate training and operation time is not cost-effective), the budget assigned here only covers the acquisition and integration of the tool into the ISE. The schedule constraints are very real and are counterproductive if too much time is spent during the selection process. Also, a development process typically yields a very costly and specific, but organically maintainable, tool. The overall constraint, apart from the actual developmental cost, is project time. The time to do the development may not be available or adequate, requiring the selection team to use a less than perfect tool, or a collection of tools to meet user requirements. In the case where tool development is the only option available, Section 5, Tool Development Process, details the process that is employed.

A tool's initial cost is the cost of purchasing it. Contributing factors include the tool's present availability and status, the time and money required for conversion and integration, and the cost of any required special equipment or support software. Cost to apply the tool include the number of applications expected and the cost of each application in terms of computer cost, labor cost, and equipment cost.

4 Tool Selection Process

A decision on whether to develop, modify, buy, or share is made prior to any procurement process but, unless pertinent information is prepared for management, nothing happens. In some cases where there is clearly an available tool that meets the defined needs and an in-depth evaluation would only serve to expend valuable resources, a tool can be selected without an extensive evaluation provided the rationale for the selection is documented. Such cases may result from tools constrained by customer requirements, by the development environment in which the IV&V activity is performed, or by situations where user experiences with particular tools outweigh the risk of selecting other tools (either due to training required, or that another tool does not perform as anticipated). The final result of this acquisition process is a list of recommended tools which address the requirements of the IV&V tasking, and meet all known user criteria. These tool recommendations are supplied to management along with pertinent related information for final decisions on tool selection. Exhibit A.4-1 illustrates the Tool Selection activities.

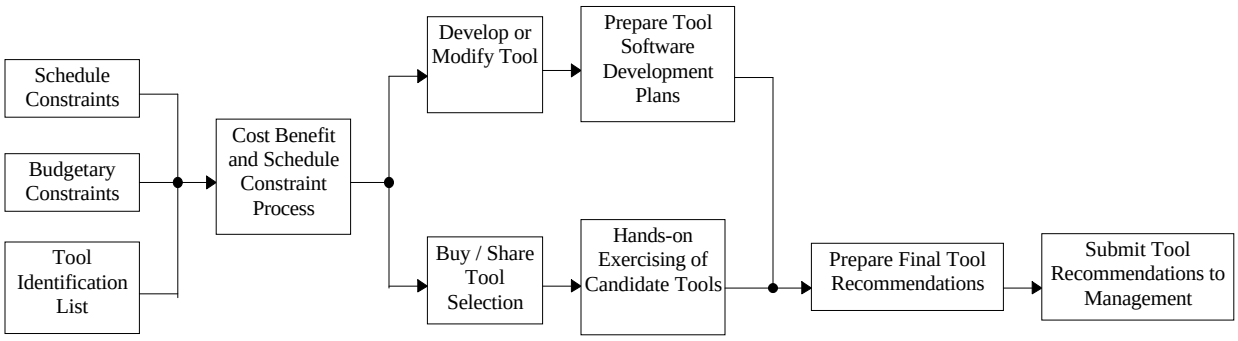


EXHIBIT A.4-1. Tool Selection Process

When tools already exist within the IV&V Program, reuse is examined first. One valuable benefit of reuse is that there may be people available that have worked with the tool, and can use the tool to its maximum potential. This is considered against potential negatives, such as how well the tool fits with other tools being used, the IV&V project requirements, and, perhaps, the limited documentation that may exist with in-house development.

If existing tools are deemed inappropriate, purchasing tools becomes the only available option. Although purchasing tools may, on the surface appear to be more expensive than reusing existing tools, one benefit of purchasing tools is that the tools can be selected to closely match project needs in terms of functional and operational requirements. The availability of training supplied by vendor and adequate user documentation, will enable the project to put the tool in place and be productive very quickly. This may lower overall cost, because higher productivity results from using a tool that closely satisfies project requirements. Additional savings that can be achieved from a COTS tool are that the more apparent defects have been identified and removed, and upgrades are much less expensive than the initial cost. A potential disadvantage of purchasing

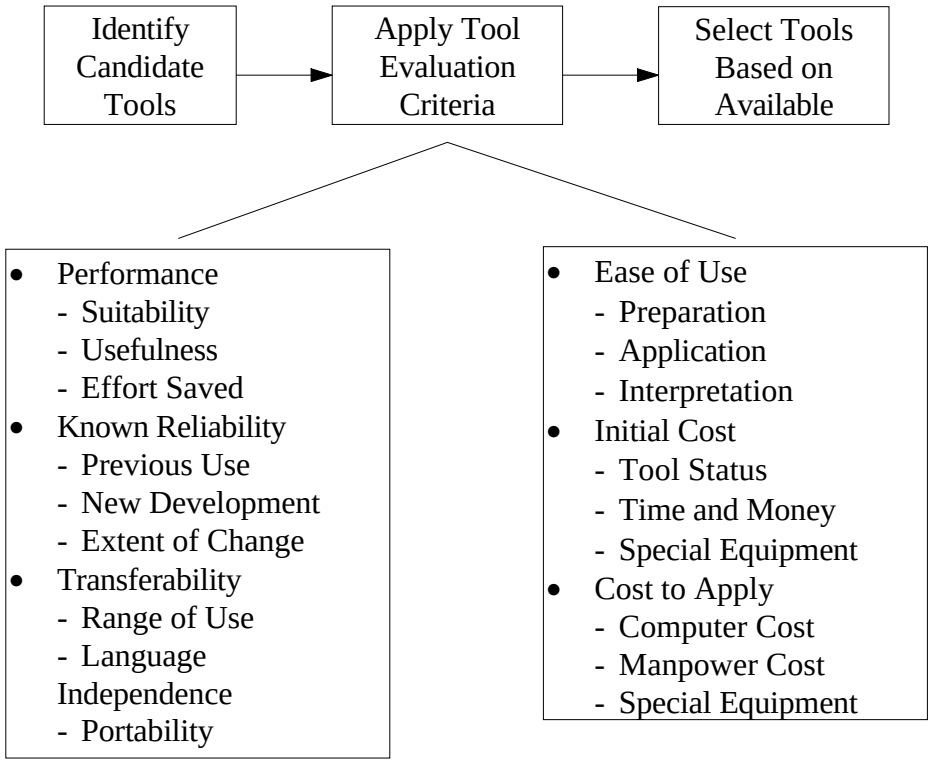


EXHIBIT A.3-4. Example Criteria and Their Related Factors

.

CRITERIA	WEIGHTING
Ease of Use	5
Intelligence/Helpfulness	5
Error Handling	5
System Interface	10
Performance	5
Robustness	15
Learnability	10
SE Environment	5
Known Reliability	5
Vendor History	5
Purchase, License or Rental Agreement	5
Maintenance Agreement	5
Training	10
Inter-Tool Interfacing	10
TOTAL	100

EXHIBIT A.3-3. Example Criteria Weighting Scheme

3.3 Tool Demonstration

One member of the tool identification team is identified as the single Point of Contact (POC) for a given tool. There are many tools, so each member may be responsible for numerous, preferably related, tools. The tools to be demonstrated are those which have been deemed acceptable by the vendor product evaluation process. The POC is responsible for contacting each tool vendor to schedule a formal demonstration. These formal demonstrations are intended for the appropriate tool identifications team members, key members from the task(s) identifying the tool requirements, and management. The team POC provides information (e.g., project context and tool requirements) to the vendor so the demonstration is targeted towards project needs. The team POC may want to send a copy of the criteria to the vendor, so the vendor can address the criteria directly. During the demonstrations, each team member should capture notes and complete the checklists (if necessary). These notes are gathered and consolidated by the team POC and placed into the tool evaluation folder

- Is the user free of all obligations to the vendor regarding use or sale of the objects generated by the tool?

Maintenance Agreement

- Does a warranty exist for the tool?
- Can the user purchase a maintenance agreement?
- Can the vendor be held liable for the malfunctioning of the tool?
- Will maintenance agreements be honored to the customer's satisfaction in the case that a vendor sells out?
- Does the maintenance agreement include copies of releases/updates?

Training

- Does the vendor provide installation support/consultation?
- Is training available?
- Is training customized for the acquiring project and individuals, with attention paid to the needs of different types of users (engineers, project managers, etc.)?
- Do the training materials or vehicles allow the user to work independently as time permits?
- Is the user provided with examples and exercises?

Data Exchange (Inter-Tool Interfacing)

Does the tool support:

- Data exchange between toolbox components?
- Data exchange between other tools?
- Known Interface Standards/Protocols?

Once all questions are answered and point values assigned, the evaluation team performs calculations to derive a ranking of prime tool candidates. Weights are assigned to categories of answers depending on the importance of each criteria or characteristic as illustrated in Exhibit A.3-3.

The identified candidate tools are evaluated using the tool evaluation criteria previously discussed. The criteria take into account both the quality and practicality of each tool and can be applied to proposed as well as existing tools. These criteria may be applied informally, or used with a point system as illustrated, in which the tools are rated numerically for each criterion and assigned a score based on the ratings for all criteria.

Tool effectiveness is the degree to which a tool can contribute to the project and is often the deciding criterion in tool selection. Contributing factors are the tool's suitability to the project, the usefulness of the results it produces, and the amount of manual effort saved through its use. Tool reliability is the extent to which the tool is known to perform its intended function consistently, and with the required precision. Factors to be considered are evidence from previous use and the extent of required changes from a known baseline. One important note is that new or highly modified tools have unknown reliability.

Tool transferability is the extent to which a tool can be used on other efforts. Contributing factors are its generality and range of use, its language independency, and its portability to different computers. Tool ease of use takes into account ease of formulating tool inputs, ease of applying the tool, and ease of interpreting the results. Exhibit A.3-4 illustrates some of the criteriathat are used to judge the various tool candidates, and the driving factors or criteria descriptors involved in making the preliminary tool selections.

- Can I/O formats be redefined (edited) by the user?
- Can the tool configuration be altered to suit user needs?

Intelligence/Helpfulness

- Is the tool interactive?
- Is quick, meaningful feedback on system status and execution progress supplied to the user?

Error Handling

- Does the tool recover from errors easily?
- Does the tool protect the user from costly errors?
- Does the tool employ periodical saving of objects to protect from data loss in the case of a system failure?

System Interface

- Does the tool provide for multiple users?
- Does the tool provide for output devices (e.g., printers)?

Performance

- If the tool supports multiple users, are response and command execution times acceptable during maximal usage?
- Can the tool running on the user's hardware, handle a development task of the size required by the user?
- Does the tool provide a mechanism to dispose of any useless by-products it generates?

Robustness

- Is the tool built in such a way that it can evolve and retain compatibility between versions?
- Can new versions of the tool interface with old versions of other related tools?
- Can new versions of the tool operate correctly on old versions of target objects?
- Can old versions of the tool operate correctly on new versions of the target objects?
- Can the tool's output be interchanged between supported hosts?

Learnability

- Does the tool simplify a problem rather than complicate it?
- Do prospective users have the background necessary to use the tool?
- Can the users operate the tool without memorizing an inordinate number of commands?
- Can the user learn a small number of simple commands initially, and gradually add more advanced commands as proficiency is developed?
- Is the tool based on a small number of easy to understand/learn concepts that are clearly explained?
- Does the tool provide the user with templates or other aids to guide interaction?
- Is there a method of using a help facility that aids the novice user by providing a step-by-step description of what to do?

Software Engineering Environment

- Does the tool run on the platform currently used by the project?
- Is installation of the tool a straightforward process?
- Can data be interchanged between the tool and other tools currently in use?

Tool History

- Does the tool have a history that indicates it is sound and mature?
- Is a complete list of all users that have purchased the tool available?

Vendor History

- Is there a staff dedicated to user support?
- Does the vendor provide a responsive, helpful hotline service?
- From consultation with others experienced with the vendor, does the vendor live up to commitments/promises?

Purchase, Licensing or Rental Agreement

- Is there a cost reduction for the purchase of multiple copies?
- Is there a corporate site license available?
- Does the user have the ability to return the tool for full refund during some well-defined, reasonable period of time?
- Is the customer given full rights and access to the source code (in the event the vendor goes out of business, no longer supports the tool, and is unable to sell off rights to the product)?

An alternative source of IV&V tools is the IV&V project at West Virginia University which is developing a methodology and associated applications that may be useful in the performance of EOSDIS IV&V tasks. Their approach is termed, "The IV&V Goal/Question/Metric" (IGQM) method. This methodology allows IV&V managers to monitor the level of risk in a software development project. Managers use historical project data as a metric to position their project relative to previous projects. Based on the amount of historical data available, managers can make decisions which other projects have made and extrapolate their own projects outcome. Historical data include the impact of decisions relative to resource allocations, schedules, costs, and tradeoffs during execution of the development effort.

Additional sources for support tools are in-house tool repositories and Commercial Off-The-Shelf (COTS). Vendor literature is readily available and easy to obtain. The identification of tools developed in-house can be difficult depending on what the application is, when the tool was developed, and who it was develop for. In-house tools, unless deliberately developed for re-use, are very specific in function, making modification of the tool software difficult since the programmers, even if they were available, may not have maintained or have access to the tool development folder from which to initiate changes.

The situation for COTS is similar to that for GOTS. However, additional agencies have collected categories of tools that could prove to be an efficient alternative source, e.g., Software Technology Support Center (STSC) tool reports, trade studies, CASE Product Guides, and the National Test Bed Software Engineering Tools Experiment Report.

The candidate list should look beyond individual tools. Specifically, when no one tool does all that is required, a set of related, compatible tools may be identified that jointly cover the requirements. The IV&V management is chartered with deciding how important it is to have all or most tools developed by the same vendor or with the same characteristics. There are great advantages in acquiring a tool set with a consistent philosophy - the burden on acquisition, training, support, and use is substantially less.

3.2 Vendor Product Evaluations

The next step in the tool identification process is the vendor product evaluations. Each prospective software tool included in the tool survey is examined for compliance with the imperative criteria. Tool Data Sheets from the previous step, are used to track the various candidates and organize the results. This is adequate to generally categorize the various tools, but it does not adequately provide for evaluation of the prospective tools. Several questions can be posed once the candidate list is shorter than initially perceived in the tool survey step. These questions in some cases and for certain applications can be very specific and can entail a weighted summation to achieve a result. Some of the example questions are listed below and will be incorporated into a response form used to formally evaluate each tool.

Ease of use

- Can various aspects of the interface be tailored to suit user needs, including application and ability level?
- Can the user define new commands or macros for commonly used command sequences?

Product Information

Product Name:_____

Version #:_____

Evaluator Information

Item Id #:

Application Class (all that apply)

_____ System Design Analysis
_____ Requirements Analysis
_____ Requirements Traceability
_____ Interface Analysis
_____ Static and Dynamic Code Analysis
_____ Software Design Analysis
_____ Communication Design Analysis
_____ Development Analysis
_____ Performance Analysis
_____ Test Preparation
_____ Test Analysis
_____ Test Execution
_____ Metrics Capturing
_____ Office Automation
_____ Configuration Management
_____ Other

Vendor Information

Name: _____

Phone: ( )
FAX: ( )

Address: _____
_____

POC: _____

Title: _____

Phone: ( )
Email: _____

Minimal Hardware Requirements

Video:
RAM:
On-Line Storage:
Off-Line Storage:

Customer Support

_____ On-Line Help
_____ Training Available
_____ Phone Technical Support
_____ Upgrades Included
_____ On-Site Technical Support

Hardware Support

_____ Macintosh
_____ 486 PC w/MS Windows
_____ 386 PC w/MS Windows
_____ Sun Sparc w/Solaris

Pricing Information

Single User: $
Site License: $
Multiple Copy: $
Maintenance Contract: $
GSA Available:

Language Support

_____ C++
_____ C
_____ Ada
_____ Other

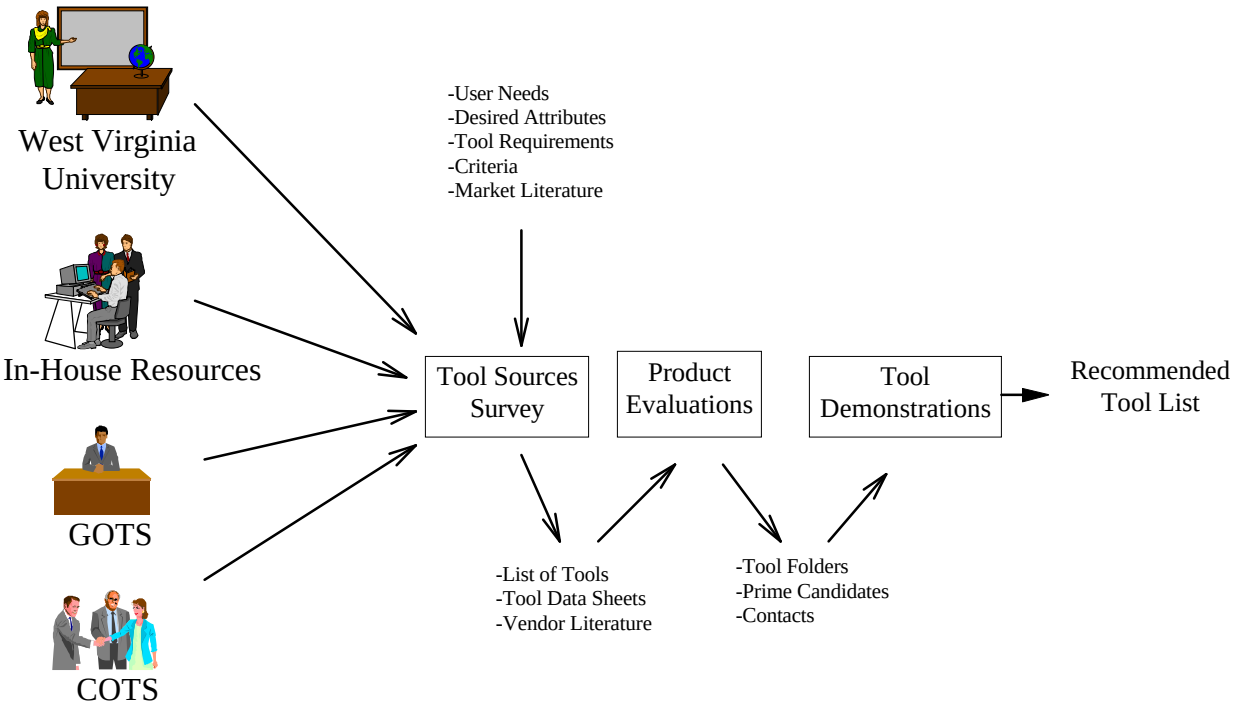
Methodologies

_____ IDEF
_____ Functional Analysis
_____ OO Analysis
_____ Structured Design
_____ OO Design

Notes

Product Description/Purpose

EXHIBIT A.3-2. Tool Data Sheet



TOOL SOURCES

EXHIBIT A.3-1. Tool Identification Activities

Software support tool documentation and literature are added to various categories of tool folders. From this literature, tools are highlighted as possible candidates for a required function or category. It is preferential to have more than one candidate to preclude any bias or favoritism. The survey process should be conducted as independently as possible. All candidate literature in the category under examination is scrutinized for general compliance with the imperative criteria for each function. Any gross deficiencies result in elimination from the survey. When the criteria are specific enough to eliminate all or most of the candidates, a tool software development process may need to be initiated. However, these criteria could be relaxed to utilize available tools, thereby avoiding significant expenses incurred by initiating a software development process. All candidate tool data is logged into a Tool Data Sheet to organize the process. An example Tool Data Sheet is given in Exhibit A.3-2.

The Tool Survey has numerous sources from which to draw. A concern is adequate support and documentation. Many tools developed for the government, and hence labeled GOTS, are one-time applications that may answer all of the imperative criteria. However, because they are one-time applications, generally they have no available follow-on support, and consequently are considered an unacceptable solution. The life cycle of the ISE is long, and adequate vendor support is a necessity. Downward compatibility is essential as the environment changes. Should downward compatibility at some time become unsupported, the vendor may supply a snapshot of his prior version, so that the tool could be tailored as necessary to maintain compliance with the ISE. From this perspective, the software would be maintained organically.

3 Tool Identification Process

Studies are conducted to identify candidate tool sources such as COTS, Government Off-The-Shelf (GOTS), public domain, West Virginia University (WVU), or in-house. A Tool Evaluation Report is generated summarizing the findings and identifying tool alternatives complying with the defined set of tool requirements and criteria.

This activity entails conducting develop/modify versus buy/share trade-off studies before committing to an in-house tool software development process. These studies assess the availability, capabilities, growth potential, costs, and platform requirements of all identified tool alternatives. Potential tools are selected and corresponding tool vendors are contacted for product information. All tools evaluated, meet the essential requirements portion of the criteria, otherwise they are removed from the candidate list. The objective is to narrow the field to allow in-depth evaluations. All findings are filed in Tool Folders for current and possible future use.

The alternative develop/modify option is always available for most tool applications, however, this tool identification process applies to only the buy/borrow/share options. The develop/modify opportunity is addressed as an option during the next section, Section 4, the Tool Selection Process. That section will assess which path will meet the needs of the user, including both time and monetary constraints. Both of these paths the develop/modify or the buy/borrow/share pass through Section 4 and proceed on to Section 5, the Tool Development/Procurement Process.

A Tool Acquisition Strategy, consistent with budget and schedule constraints, is devised. Candidate tool vendors are contacted and requested to provide a demonstration of their product. These formal demonstrations are intended to present compliance with the specific task requirements and previously established tool criteria. Any tool failing to meet all reasonable requirements and criteria is removed, and the remaining candidates are requested to supply demonstration copies for hands-on execution by an evaluation team.

Exhibit A.3-1 illustrates the various activities involved in the tool identification process that begins with the tool sources survey, and culminates with a tool selection list that is ultimately passed to the tool selection process.

3.1 Tool Survey

The goal of the Tool Survey is the identification of candidate tools that address perceived needs of the user. It is not the charter of the Tool Survey to perform in-depth analysis or execution of the tool software. The product of the Tool Survey is a list of tools that meet at least a majority of the imperative criteria, with comments on how each tool addresses the non-imperative criteria, and additional comments that may be pertinent during the formal selection process.

Tool Function or Need	Requirement or Critical Criteria	Non-Critical Criteria
Configuration Management	• Maintain CM of ISE Elements • Provide CM at the File Level • Support Baseline Creation • Support Code, Graphics, Text, Tests, etc.	• Ease of Use • Cost • Ease of Insertion • Quality of Support
Project Management Tool	• Export Data to Artemis Prestige • Produce PERT and Gannt Charts • Baselining Schedule Data for Plan Versus Actual Tracking • Scheduling Tasks Based upon Level of Effort or Duration	• Ease of Use • Cost • Functionality • Vendor Support • Robustness
Desktop Publishing	• Executes under Solaris • Includes Conversion Capabilities to Rich Text Format • Support Text/Graphic Integration • Supports Multi-file Documents • Provides On-line Templates • Provides Variable Definition Capabilities • Provides File cross-reference Capabilities • Supports Table Creating and Modification	• Ease of Use • Vendor Support • Cost • Availability
Word Processing	• Executes in the PC Windows and Macintosh Environments • Imports and Exports data in Various Formats • Supports Multi-file Documents • Provides On-line Templates	• Ease of Use • Minimal Training Needed • Vendor Support • Data Exchange
Network Communications	• Compatible with PC Windows Environment • Compliant with LAN and WAN Integration	• Ease of Use • Expandable • Cost

EXHIBIT A.2-3. Examples of Critical and Non-Critical Criteria

ISE Tool Requirement Numbers	REQUIREMENTS
ISE-T4-0010	The ISE shall include a Computer-Aided Software Engineering (CASE) tool for: •Graphically representing analysis and design models, and •Defining and generating database schemas.
ISE-T4-0020	The ISE shall enable the generation of plans and procedures for developing the ISE.
ISE-T4-0030	The ISE shall enable the generation of ISE development schedules.
ISE-T4-0040	The ISE shall enable the generation of ISE status reports.
ISE-T4-0050	The ISE shall include a tool for maintaining configuration control of ISE design and development modules.
ISE-T4-0060	The ISE shall enable the creation of ISE review presentation material.
ISE-T4-0070	The ISE shall include a client/server development tool to promote information exchange.
ISE-T4-0080	The ISE shall include issue tracking capabilities used for tracking ISE development issues.
ISE-T4-0090	The ISE shall enable access to ISE training guidelines and material.
ISE-T4-0100	The ISE shall enable the production of ISE requirements, design, test, and user's guide documents.
ISE-T4-0110	The ISE shall include the development tools necessary to build applications to satisfy end user needs.

EXHIBIT A.2-2. ISE Tool Requirements

The imperative and non-imperative criteria are the definitive aspects of tool assessment. Each tool addresses a requirement or needed function. In turn each tool has numerous criteria used to evaluate it. The imperative criteria address the initial requirements where the non-imperative weighted criteria address many of the operational and feature aspects of the tool in question. Exhibit A.2-3 identifies several basic IV&V functions requiring tools and some example criteria which could be used to evaluate them.

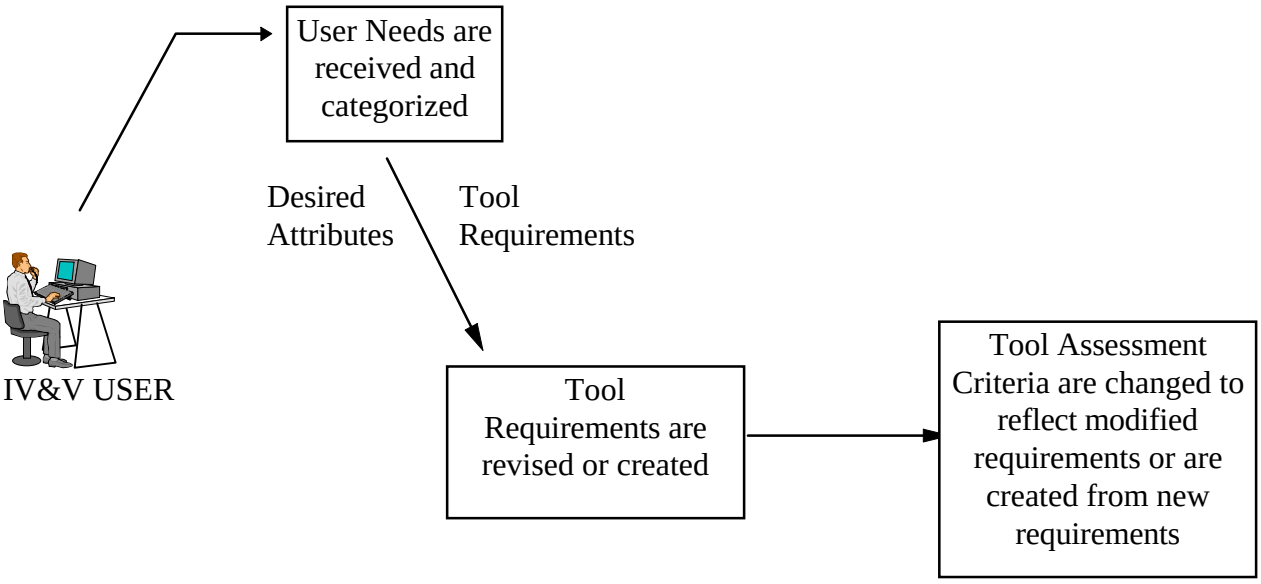


EXHIBIT A.2-1. Requirements Definition Process

and requirements traceability tool. Other relevant user requirements may include categories such as platforms, compatibility with other tools, urgency and degree of complexity. Some associated attributes include user friendly, small learning curve, and interactive capability. The tool identification team captures these needs, separates the requirements from the desired attributes, and formally documents each. Both the requirements and the desired attributes are input to the development of tool assessment criteria.

Several requirements detailing needs are already imposed upon the ISE so that the IV&V infrastructure can be successfully established. These requirements, which are added to and modified as user needs are addressed, are summarized in Exhibit A.2-2 and detailed in the ISE IV&V System Requirements Document.

2.3 Tool Assessment Criteria

The tool requirements and attributes, created by assessments of user needs and the identification of user requirements, are utilized to develop the tool assessment criteria. The list of criteria is divided into two categories: imperative and non-imperative. The imperative criteria are the minimum requirements mandated to satisfy user needs. The non-imperative criteria are means to measure tools against each other. This latter group includes less tangible criteria such as learning curve, user friendliness, product support, and training. These criteria have weights assigned to them to help differentiate the tools, whereas, the imperative criteria do not have weights since each is a hard, fast requirement and failure to meet one such criteria disqualifies a tool. Numerous technical interchange meetings and conversations are required to clearly identify the deciding criteria upon which the tool selections are based. Establishing robust criteria is most important as it is used throughout the remainder of the process.

2 Requirements Definition Process

The requirements definition process begins with a needs assessment study and tool requirements analysis, drawing upon:

- Ancillary tool requirements as dictated by the ISE and the existing tool infrastructure,
- The IV&V Plans currently in place,
- IV&V task activities (current and planned), and
- New or revised requirements flowed up from the existing tool suite (e.g., user feedback).

During this phase, the tool needs are assessed and user scenarios and requirements are developed. Useful tool capabilities and the specific objectives that each tool should satisfy are identified. The specific IV&V task requirements, scheduled need dates, and task loading are evaluated to determine where high tool payoff potentials exist. The results of this requirements definition process are documented in a Tool Requirements Specification.

2.1 Needs Assessment

All tool procurement, developed or COTS, begins with an assessment of the user's needs. If there is no need identified by the user, there is no requirement for a tool. The user is the driving force behind tool definition. This process begins with an assessment of the needs of the users. When the ISE is in place, a concurrent feedback process will actively poll users for current tool satisfaction. A number of things can happen. The current tool suite as integrated into the ISE is not adequate to address some current users needs, or a tool previously introduced to the community does not adequately respond to user needs. In either case, a tool will be modified or purchased (e.g., COTS or GOTS tools).

User needs are collected in several ways:

- Informally, by a telephone conversation (which generates a contact form to be filed against a specific tool in its Tool Folder or into the ISE Tool Requirements Folder),
- By completing and returning a user feedback form, or
- By logging into the Tool Box and creating a tool deficiency report.

In all cases a need is identified to the ISE maintainers. Exhibit A.2-1 illustrates the Requirements Definition Process.

2.2 Tool Requirements Identification

The users' needs have been captured as indicated in Section 2.1. Next, the tool requirements are categorized and identified. Categories of requirements include tool types such as word processor

issues CM control version numbers and maintains version control for subsequent changes. The tool next enters an Integration activity and is released to the ISE users. The concluding process or activity ensuring the tool meets the users needs is the user feedback process. This process supplies a "satisfaction index" to the developers/maintainers of the tool in question. Through this process the tool requirements are modified based on user needs and recommendations. This in turn drives a reassessment and a decision to select a new COTS tool or modify the existing tool.

managers. The Test Buddy comes equipped with a modem and communication software enabling remote users to upload and download information and data to/from the IIR.

1.2 Tool Management Overview

The goal of tool management in support of maintenance and operation, is to provide a set of tools that operate properly, and to protect the tools as NASA assets. It is important to focus the ownership of the tool repository on one individual or team. It is the responsibility of the manager of the tool repository to make sure that the tools operate properly, that the users have the latest releases, and that deficiencies are reported and fixed. Tool library and repository management provide a focal point and mechanism for tool maintenance.

Exhibit A.1-5 illustrates the top level activities involved in the tool management process of the ISE. The initial management aspects of tool selection/development involve the identification of requirements and the creation of criteria to judge the approach needed to integrate a tool into the ISE. Some subsequent aspects vary depending upon whether the tool is developed/modified vice bought/shared/borrowed. All tools, in either category, are exercised to verify compliance with the tool requirements and acceptance criteria. Testing differs for each category of tool, but in both cases is iterative. Developmental testing begins at the module level, whereas testing for a purchased or shared tool begins at the program level, because no lower level component visibility is available. To resolve any tool deficiencies, appropriate changes can be made to a developed tool and retested, or in the case of Commercial Off-The-Shelf (COTS) tools, another vendor's product may satisfy requirements more comprehensively.

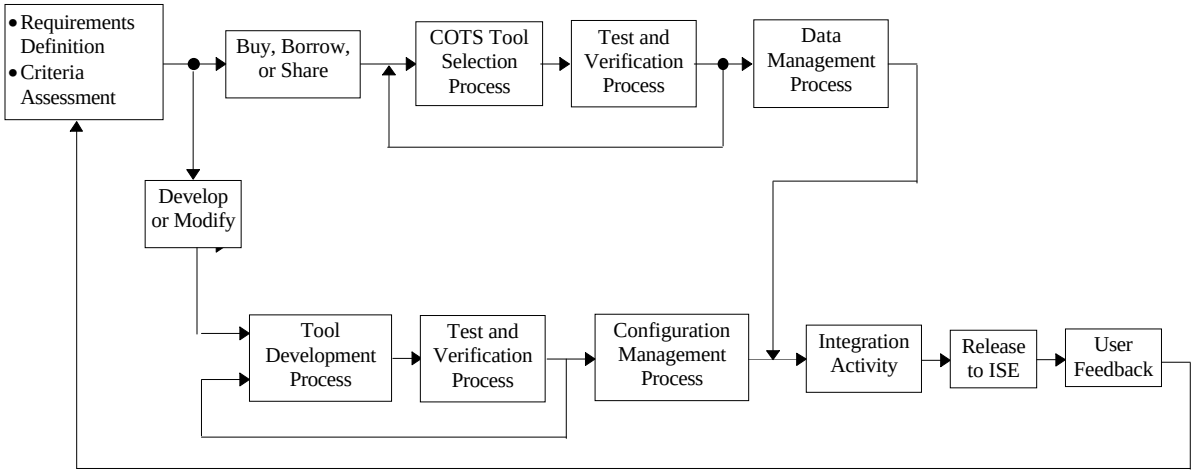


EXHIBIT A.1-5. Tool Management Processes Overview

Each COTS tool enters a Data Management process, identifying the software and all of its purchased/borrowed elements. While, developed software follows a Configuration Management (CM) path. This developmental process not only identifies all of the elements of the tool, but

- Formal Documents (e.g., IV&V Management Plans, Certification Plans, Test Master Plans),
- Reports (e.g., Analysis Reports, White Papers, Discrepancy Reports),
- Presentations (e.g., Status Reviews, NASA Briefings),
- Project Status (e.g., Schedules, Project Risks, Management Reports, Staffing Profiles, DR Status), and
- Test Documentation (e.g., Test Plans/Procedures, Test Scenarios, Test Results, Post Test Analysis).

IV&V data products consist of:

- Requirements Traceability Data, and
- Performance Metrics.

The final type of information located in the IIR is NASA support material. This is documentation (e.g., standards, guidelines) that will be placed on-line to expedite the IV&V process. Because of the variety of information that exists in the IIR, not all users have access to all information. Security exists, and access is provided on a need-to-know basis. Although the type of user matters, the location of the user does not. The IIR, located at the West Virginia University (WVU)/NASA Software IV&V Center in Fairmont, West Virginia, is accessible to the IV&V user community, regardless of their location, via the established communication architecture. The user interface to the IIR is provided by front-end tools located in the ISE Toolbox.

The Toolbox compartmentalizes all of the tools to meet the following goals:

- Allow Access to the IIR,
- Support Daily IV&V Activities, and
- Establish and Maintain the ISE.

Front-end tools enable users to access the data and products stored in the IIR while providing a standard look and feel for navigating through the IIR. The Toolbox also contains office support tools, project management tools, test tools, performance and code analyzing tools, and interface analysis tools to support the daily needs of the IV&V task members. Where feasible, these tools are integrated to facilitate the sharing of data, enabling information to progress through the IV&V Program life cycle. Finally, the Toolbox houses the tools necessary for the ISE support staff to establish and maintain the ISE. These tools are used to allow ISE support staff to develop new tools, configure a requirements traceability database, and perform configuration management.

The third and final component of the ISE, the Test Buddy, is a portable testing platform that enables IV&V task team members to perform remote site testing. The Test Buddy is self-contained, housing all of the tools required for remote users to conduct tests, capture the results, and rapidly distribute debriefing messages and "quick-look" reports directly to IV&V and NASA

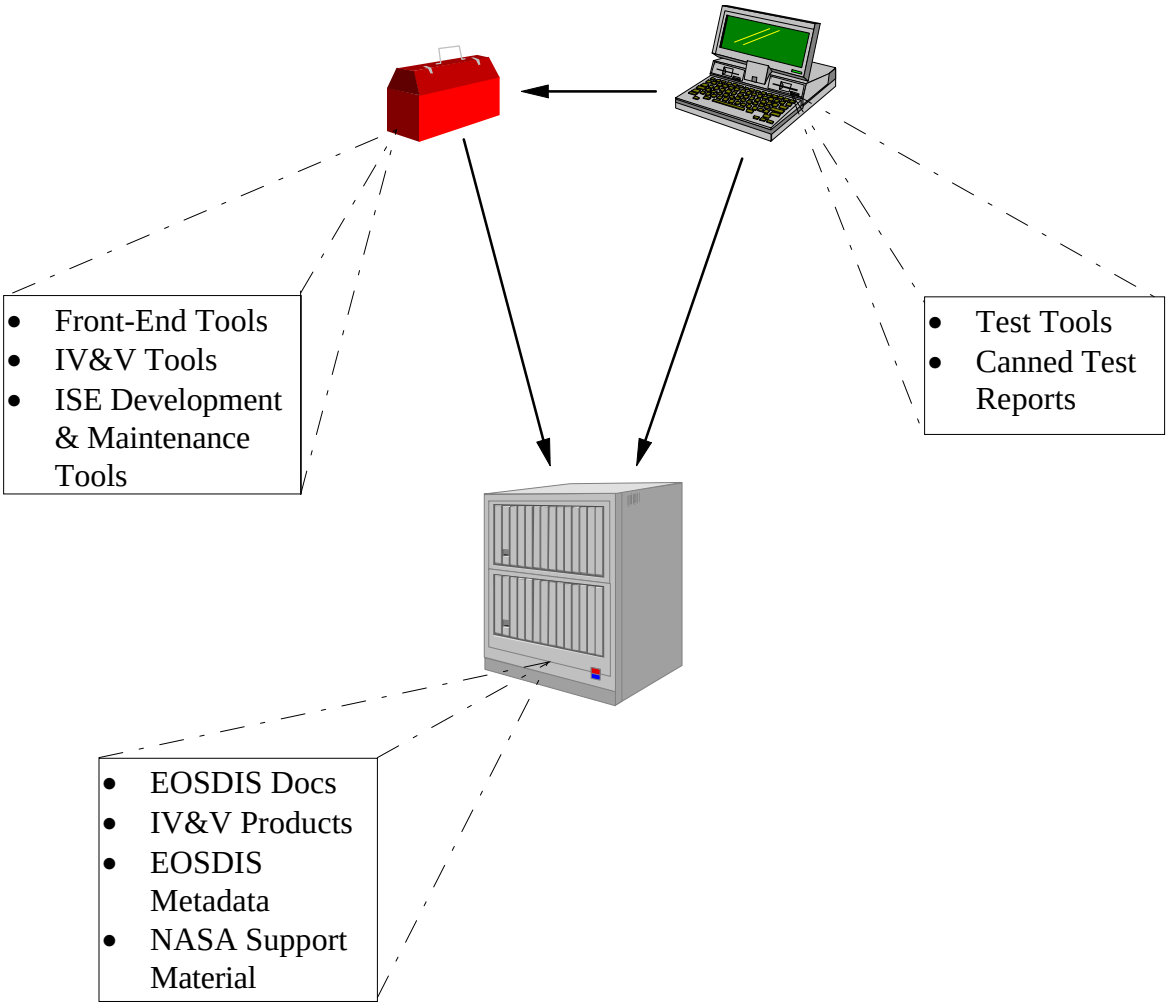


EXHIBIT A.1-4. The Integrated Support Environment

- EOSDIS Metadata,
- EOSDIS IV&V Products, and
- NASA Support Material.

By storing this information, the IIR assists the entire user community in their daily activities. The EOSDIS development and integration documentation provide IV&V analysts with on-line, up-to-date EOSDIS documentation pertinent to the IV&V effort (e.g., requirements, design and implementation specifications, system design and development metrics, management and test plans, schedules and other reference material). In cases where the documentation cannot be retained on-line, metadata directs the user to where the documentation can be found.

The IV&V products that are stored in the IIR fall under two categories: information and data. IV&V informational products consist of:

- Provide the remote tester with the capability to submit test results quickly and efficiently,
- Have the capability to access information stored in the IIR, and
- Be flexible enough to support changes in test scenarios due to unforeseen circumstances.

The ISE objectives were developed to be responsive to the following system level operational requirements depicted in Exhibit A.1-3.

EOSDIS Operational Requirement Number	ISE Requirement
ISE-OPS-0010	The ISE shall be accessible to geographically dispersed users via modem or Wide Area Network (WAN).
ISE-OPS-0020	The ISE shall provide capabilities for the establishment of user accounts.
ISE-OPS-0030	The ISE shall perform user authentication upon user login.
ISE-OPS-0040	The ISE shall implement user access control based upon user login.
ISE-OPS-0050	The ISE shall provide a tailored Graphical User Interface (GUI) for accessing tools and data that is based upon user login information.
ISE-OPS-0060	The ISE shall allow data transfer to geographically dispersed users.

EXHIBIT A.1-3. System Level ISE Operational Requirements

The IIR is the heart of the ISE, holding all the IV&V project data and results as well as the EOSDIS development information. This repository is accessible to the user community via front-end tools provided by the Toolbox. Along with the front-end tools, the Toolbox contains all of the IV&V tools necessary for the IV&V task team to perform their daily tasks. The output products and data generated or utilized by the tools are stored in the IIR.

The third and final component, the Test Buddy, is a self-contained, portable testing platform. The Test Buddy enables IV&V testers to complete test scenarios and cases on site, with the ability to access various data and information repositories (e.g., procedures, check lists for conducting tests, previous test case results, configuration identification) in the IIR, and to rapidly submit test results. Exhibit A.1-4 depicts these three components of the ISE and their relationships.

The IIR serves a role as a database librarian, storing:

- EOSDIS Development and Integration documentation (including Configuration ID, Schedules, Software Problem Reports (SPR), Change Requests (CRs)),

community to view, and provided they have the proper access, create and update. ISE tools support the following minimal set of capabilities:

- System Design Analysis
- Requirements Analysis
- Requirements Traceability
- Interface Analysis
- Static and Dynamic Code Analysis
- Software Design Analysis
- Communication Design Analysis
- Development Analysis
- Performance Analysis
- Test Preparation
- Test Analysis
- Test Execution
- Discrepancy Tracking and Reporting
- Metrics Capturing
- Office Automation
- Configuration Management

Functionally, the second and third objectives are combined requiring the ISE to "provide visibility to EOSDIS development and integration documentation" and "provide visibility to EOSDIS IV&V products." A wide multitude of users, from a selected set of locations, are accessing the ISE to view the information stored in the IIR. In order to support this diversity of users and their needs, the ISE is directed to:

- Provide geographically dispersed users access to this information,
- Provide centralized control over the information,
- Provide accessibility via the Internet, modem, or through the NASA Wide-Area Network (WAN),
- Provide a standard user interface to access this information, and
- Provide Security, supplying access to data as the user's needs require.

Finally, the ISE is to "provide remote site IV&V testing capability". This is supported mainly through the Test Buddy, a stand-alone, portable testing platform. However, just providing the platform is not enough. The ISE, specifically the Test Buddy, must:

1.1 Tool Infrastructure

The preface to this appendix identifies three discrete components composing the ISE. Together these components support the user community by meeting the objectives as identified in Exhibit A.1-2.

OBJECTIVES	COMPONENT	METHOD
• Provide required IV&V tool support	Toolbox	• The Toolbox houses IV&V tools to support daily needs
• Provide visibility to EOSDIS development and integration documentation	Integrated Information Repository & Toolbox	• The IIR archives all EOSDIS products and data relevant to the IV&V effort • The Toolbox provides front-end tools to access these products and data
• Provide visibility to EOSDIS IV&V products	Integrated Information Repository & Toolbox	• The IIR archives all EOSDIS IV&V products and data • The Toolbox provides front-end tools to access these products and data
• Provide remote site access to IV&V test information	Test Buddy	• The Test Buddy is a standalone, portable platform

EXHIBIT A.1-2. ISE Objective/Component Mapping

The capabilities and characteristics of the ISE needed to meet these objectives are being defined. The paramount objective of the ISE is "provide required IV&V tool support." The tools to meet this objective reside in the ISE's Toolbox, and are robust enough to support the IV&V effort in day-to-day activities. Products from this effort are being placed into the IIR for the entire user

1 Tool Management Approach

This appendix details the approach for managing the Integrated Support Environment (ISE) toolset for Earth Observation System Data and Information System (EOSDIS) IV&V. The purpose of the ISE is to provide consistent, accessible, and controlled use and availability of tools and information to the ISE user community, independent of user location or source of information. The ISE provides visibility to EOSDIS system development and integration, and associated IV&V products. The environment provides real-time remote site testing, plus the flexibility to keep pace with technology. The ISE consists of an Integrated Information Repository (IIR), a Toolbox, and a Test Buddy. The ISE supports a user community consisting of ISE support staff, IV&V task teams, NASA project level users, EOSDIS developers, and the scientific user community. The hub of the ISE is the IIR that holds IV&V project data and EOSDIS development information. The Toolbox supplies the user community access to the IIR with an array of front end access tools. The Test Buddy is a self-contained, portable platform that has direct access to the IIR and is used to test related information. Additional details concerning the ISE are found in the ISE EOSDIS IV&V System Requirements document.

Management of the ISE and its resident tools is in accordance with procedures detailed in this plan. These procedures encompass specifying, acquiring, integrating, testing, training, operating and maintaining the ISE tool suite. The methods used to manage the IV&V tools are responsive to the needs of the project, result in cost effective solutions, and provide for the qualification and control of the tools necessary to ensure the integrity of tool products. Exhibit A.1.0-1 illustrates the overall tool management activity network.

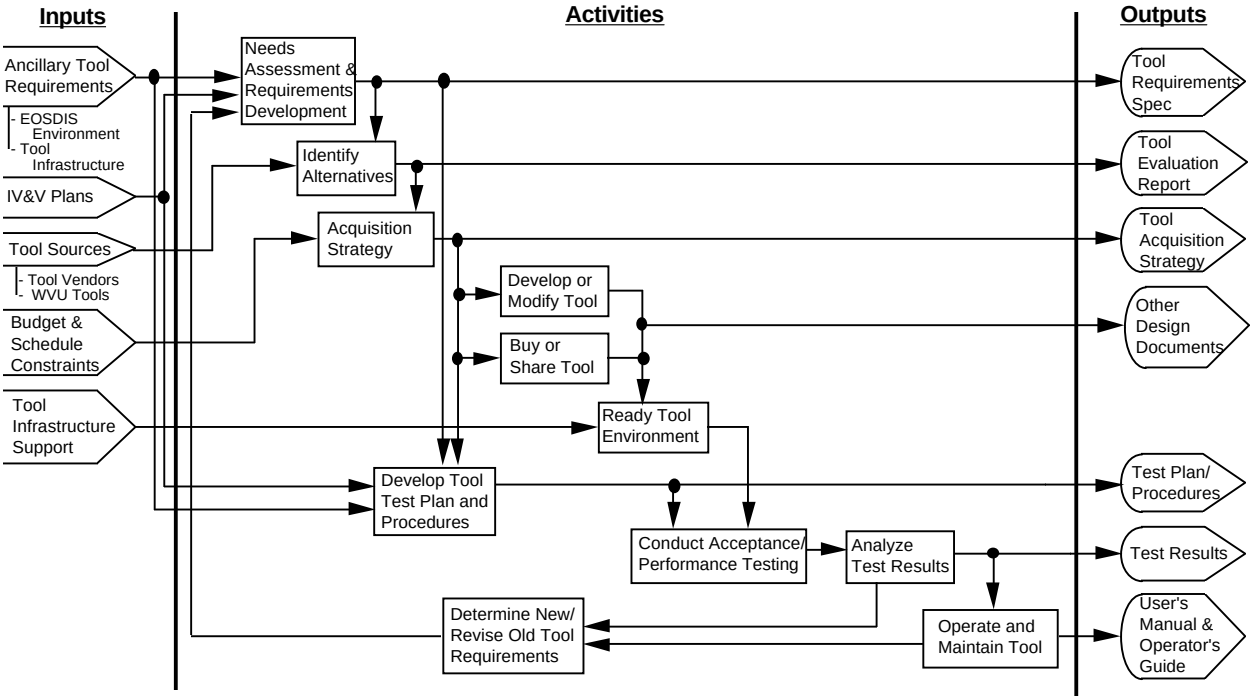


EXHIBIT A.1-1. Tool Acquisition, Development and Maintenance

LIST OF EXHIBITS

EXHIBIT A.1-1. Tool Acquisition, Development and Maintenance..... A.1-1

EXHIBIT A.1-2. ISE Objective/Component Mapping..... A.1-2

EXHIBIT A.1-3. System Level ISE Operational Requirements..... A.1-4

EXHIBIT A.1-4. The Integrated Support Environment A.1-5

EXHIBIT A.1-5. Tool Management Processes Overview..... A.1-7

EXHIBIT A.2-1. Requirements Definition Process..... A.2-2

EXHIBIT A.2-2. ISE Tool Requirements..... A.2-3

EXHIBIT A.2-3. Examples of Critical and Non-Critical Criteria..... A.2-4

EXHIBIT A.3-1. Tool Identification Activities..... A.3-2

EXHIBIT A.3-2. Tool Data Sheet..... A.3-3

EXHIBIT A.3-3. Example Criteria Weighting Scheme..... A.3-7

EXHIBIT A.3-4. Example Criteria and Their Related Factors..... A.3-8

EXHIBIT A.4-1. Tool Selection Process..... A.4-1

EXHIBIT A.5-1. Tool Development Process A.5-2

EXHIBIT A.6-1. Data and Configuration Control A.6-1

EXHIBIT A.6-2. Card Catalog Entry Log Form..... A.6-4

EXHIBIT A.6-3. Equipment Item Entry Log Form A.6-5

EXHIBIT A.6-4. Notebook Checkout Card A.6-6

EXHIBIT A.6-5. Relationship of SPR and CR Life cycles..... A.6-8

EXHIBIT A.6-6. IV&V SPR Process..... A.6-8

EXHIBIT A.6-7. Example of SPR..... A.6-9

6.1.1 Data Management Control....6-3
6.1.2 Configuration Management Control....6-3
6.2 Acceptance/Validation Testing....6-11
6.3 Tool Validation and Post Test Analysis....6-11
7 Training and Operation....7-1
7.1 User Training....7-1
7.2 Tool Implementation/Utilization....7-2
8 Life Cycle Maintenance....8-1
8.1 Tool Maintenance Improvement Process....8-1
8.2 Tool Modification/Upgrade/Substitution....8-2
9 Glossary and List of Acronyms/Abbreviations....9-1
9.1 Glossary....9-1
9.2 List of Acronyms/Abbreviations....9-3
10 References....10-1

TABLE OF CONTENTS

Section ***Page***

1 Tool Management Approach 1-1

1.1 Tool Infrastructure 1-2

1.2 Tool Management Overview 1-7

2 Requirements Definition Process 2-1

2.1 Needs Assessment 2-1

2.2 Tool Requirements Identification 2-1

2.3 Tool Assessment Criteria 2-2

3 Tool Identification Process 3-1

3.1 Tool Survey 3-1

3.2 Vendor Product Evaluations 3-4

3.3 Tool Demonstration 3-7

4 Tool Selection Process 4-1

4.1 Cost Benefit/Schedule Constraints Analysis 4-2

4.2 Tool Evaluation Exercise 4-3

4.3 Final Tool Recommendations 4-3

4.4 Tool Selection by Management 4-3

5 Tool Development/Procurement Process 5-1

5.1 Tool Development/Procurement Plans 5-1

5.2 Tool Environment Preparation and Integration Plan 5-3

5.3 Test Plans & Procedures 5-4

6 Tool Configuration Management (CM) Process 6-1

6.1 Configuration and Data Management Control 6-2

IVVMP Appendix A:
EOSDIS IV&V TOOL MANAGEMENT PLAN

(Deliverable 0301 App A)

December 2, 1994

PREPARED BY:

_____

Timothy L. Davis
Technical Lead

REVIEWED BY:

_____

Richard A. Rogers
Task Leader

RECEIVED BY:

_____

Lee LaCoste
Document Log Manager

APPROVED BY:

_____

Ron Cariola
Program Manager

Prepared By:

Intermetrics Systems Services

WVU/NASA Software IV&V Facility

100 University Drive

Fairmont, West Virginia 26554

IVVMP Appendix A:
EOSDIS IV&V TOOL MANAGEMENT PLAN

(Deliverable 0301 App A)

December 2, 1994

Prepared By:

Intermetrics Systems Services

WVU/NASA Software IV&V Facility

100 University Drive

Fairmont, West Virginia 26554

Prepared For:

NASA Goddard Space Flight Center

EOSDIS Project, Code 505
Greenbelt, Maryland 20770